```
#include <stdio.h>
#include <string.h>

#define MAX_FILES 100
#define NAME_LEN 30

struct Directory {
    char fileNames[MAX_FILES][NAME_LEN];
    int fileCount;
};

void createFile(struct Directory *dir, const char *name) {
    // Check for duplicate file
    for (int i = 0; i < dir->fileCount; i++) {
        if (strcmp(dir->fileNames[i], name) == 0) {
            printf("Error: File '%s' already exists!\n", name);
            return;
        }
    }
    if (dir->fileCount < MAX_FILES) {
        strcpy(dir->fileNames[dir->fileCount], name);
        dir->fileCount++;
        printf("File '%s' created successfully.\n", name);
    } else {
        printf("Error: Directory is full!\n");
    }
}
```

```
void deleteFile(struct Directory *dir, const char *name) {
    int found = 0;
    for (int i = 0; i < dir->fileCount; i++) {
        if (strcmp(dir->fileNames[i], name) == 0) {
            found = 1;
            for (int j = i; j < dir->fileCount - 1; j++) {
                strcpy(dir->fileNames[j], dir->fileNames[j + 1]);
            }
            dir->fileCount--;
            printf("File '%s' deleted successfully.\n", name);
            break;
        }
    }
    if (!found) {
        printf("Error: File '%s' not found!\n", name);
    }
}

void searchFile(struct Directory *dir, const char *name) {
    for (int i = 0; i < dir->fileCount; i++) {
        if (strcmp(dir->fileNames[i], name) == 0) {
            printf("File '%s' found at position %d.\n", name, i + 1);
            return;
        }
    }
    printf("File '%s' not found.\n", name);
}
```

```
void listFiles(struct Directory *dir) {
    if (dir->fileCount == 0) {
        printf("Directory is empty.\n");
    } else {
        printf("Files in Directory:\n");
        for (int i = 0; i < dir->fileCount; i++) {
            printf("%d. %s\n", i + 1, dir->fileNames[i]);
        }
    }
}

int main() {
    struct Directory dir;
    dir.fileCount = 0;

    int choice;
    char name[NAME_LEN];

    while (1) {
        printf("\nSingle-Level Directory Operations:\n");
        printf("1. Create File\n2. Delete File\n3. Search File\n4. List Files\n5. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                printf("Enter file name to create: ");
                scanf("%s", name);
```

```
            createFile(&dir, name);
            break;
        case 2:
            printf("Enter file name to delete: ");
            scanf("%s", name);
            deleteFile(&dir, name);
            break;
        case 3:
            printf("Enter file name to search: ");
            scanf("%s", name);
            searchFile(&dir, name);
            break;
        case 4:
            listFiles(&dir);
            break;
        case 5:
            printf("Exiting...\n");
            return 0;
        default:
            printf("Invalid choice! Try again.\n");
      }
   }

   return 0;
}
```

```
Single-Level Directory Operations:
1. Create File
2. Delete File
3. Search File
4. List Files
5. Exit
Enter your choice: 1
Enter file name to create: ABC
File 'ABC' created successfully.

Single-Level Directory Operations:
1. Create File
2. Delete File
3. Search File
4. List Files
5. Exit
Enter your choice: 2
Enter file name to delete: ABC
File 'ABC' deleted successfully.
```

```
Single-Level Directory Operations:
1. Create File
2. Delete File
3. Search File
4. List Files
5. Exit
Enter your choice: 4
Directory is empty.

Single-Level Directory Operations:
1. Create File
2. Delete File
3. Search File
4. List Files
5. Exit
Enter your choice: 5
Exiting...


...Program finished with exit code 0
Press ENTER to exit console.
```